

**Описание процессов, обеспечивающих поддержание жизненного цикла  
программы для ЭВМ «Insiware CoMForT»**

# 1. Процесс работы с дефектами ПО

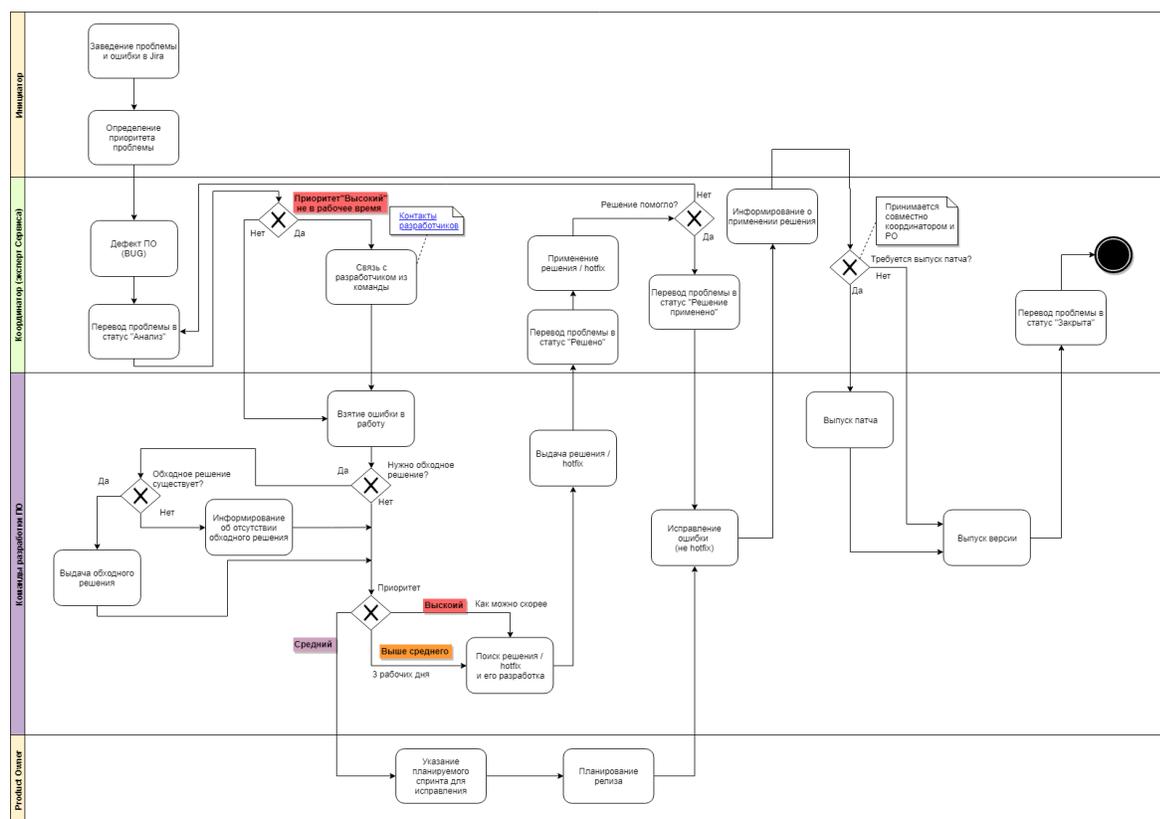


Рисунок 1. Схема взаимодействия

## Дополнительная информация:

- Правила определения приоритета проблем и ошибок, сроки исправления.
- Работа с проблемами и ошибками среднего приоритета.
- Инициатор и координатор проблемы могут быть одним и тем же человеком.
- Обходное решение – это действия, которые нужно предпринять, чтобы избежать повторения проблемы за счет дополнительных действий. Обходное решение может выдать как координатор, так и разработка. Если обходное решение нужно получить от разработки, то координатор дополнительно указывает это в Jira. Обходного решения может не существовать.
- Hotfix - решение (чаще всего jar-файл или sql-скрипт), которое выдает разработка. После применения hotfix обходное решение применять не надо, и проблема становится решенной. Hotfix отличается от полноценного решения тем, что полноценное решение оформлено в виде патча (или версии).

2. Работа в модуле управления проблем PRB
  - 2.1. Инженер команды HelpDesk (далее HD) регистрирует проблему в JIRA с типом задачи = "bug". Инженер является координатором проблемы и производит сбор всей необходимой информации для передачи в группу разработки.
  - 2.2. После сбора информации инженер HD передает проблему координатору. Координатор проблемы:
    - Может запросить дополнительный анализ по проблеме;
    - Может разделить проблему на несколько, если это выяснится в ходе анализа;
    - Выдает обходное решение (workaround) по проблеме в зависимости от степени критичности (см. таблицу приоритетов дефектов софта);
    - Выдает решение проблемы в виде номера версии, в которой будет включено исправление проблемы, а также срок выдачи этого исправления.
  - 2.3. После выдачи решения, координатором проблемы становится инженер, который зарегистрировал проблему.
  - 2.4. Координатор контролирует применение выданного решения, а также статус исправления проблемы:
    - В случае, если выданное решение не помогло – координатор проблемы возвращает проблему в группу разработки с подробным описанием последствий применения решения.
    - В случае, если выданное решение устранило проблему – координатор, закрывает проблему в JIRA.

### 3. Планирование ошибок среднего приоритета

#### 3.1. Общие принципы

Ошибки приоритета Major не являются критичным для продукта и процессов заказчика и должны планироваться разработкой в нормальном режиме. Однако, планированию таких ошибок нужно уделять должное внимание и делать это на регулярной основе, так как эти ошибки сказываются на работе наших пользователей, влияют на восприятие наших продуктов в целом и дают возможность нашим конкурентам вставлять нам шпильку в самые неудобные места.

#### 3.2. Сейчас выделяются 2 класса ошибок:

- Ошибки, обнаруженные нашими клиентами, по которым есть заведенные проблемы в JIRA. Планированию этих ошибок нужно уделять повышенное внимание, так как это напрямую влияет на количество заявок, нагрузку на сервис, а иногда и на разработку.
- Ошибки, о которых клиенты еще не знают, определенные в рамках внутренних процессов компании.

#### 3.3. Планирование исправлений ошибок, привязанных к проблемам

Процесс работы с ошибками может отличаться в командах, но базовый цикл должен быть примерно следующим:

- В течение заданного SLA (2 недели) новая ошибка должна быть запланирована к исправлению на конкретный месяц разработки, для этого
  - Ошибка разбирается PO и Team Lead с точки зрения внятности описания и достаточности материалов
  - PO должен подтвердить реальный приоритет для ошибки. У нас нет цели исправлять все ошибки, это в принципе сейчас невозможно. Изменение приоритета должно быть согласовано с ответственным (Service Manager, Project Manager), в случае конфликта PO должен выносить проблему на встречу по качеству с менеджером по качеству
  - PO и Team Lead должны оценить трудозатраты по исправлению ошибки, обозначить возможные границы и риски.

- После первоначальной модерации ошибки и ее оценки РО планирует ошибку на конкретный месяц разработки с учетом загрузки команды.
- При планировании конкретных релизов РО учитывает бэклог ошибок, запланированных к исправлению на конкретный месяц, и планирует исправление в релиз исходя из загрузки команды. Эти планы обсуждаются на встречах по качеству с менеджером по качеству
- Переносы сроков исправлений ошибок должны согласовываться со стейкхолдерами.

#### 4. Процесс работы с ошибками среднего приоритета

4.1. Проблемы и ошибки с приоритетами critical, blocker, project blocker не планируются, а решаются как срочные согласно определенным уровням SLA.

#### 4.2. Заведение ошибки и проблемы

Для того, чтобы ошибка среднего приоритета была исправлена необходимо:

- наличие проблемы с приоритетом Серьезный (Major) в статусе Анализ
- наличие ошибки в Jira

Если требуется исправить ошибку в какой-то определенный срок (например, в сентябре 2025 года), то необходимо это указать в поле planned fix date. Если срок не указан, то ошибка планируется в первый свободный месяц.

Заведенная ошибка должна быть в статусе Backlog.

### 4.3. Анализ Product Owner и планирование

На основании описания и желаемой даты исправления, Product Owner (далее PO) команды в течение 10 рабочих дней либо:

- устанавливает ошибке месяц/версию исправления и переводит ошибку в статус ToDo
- отклоняет/уточняет ошибку (снижает приоритет, закрывает, задает вопросы) с соответствующим комментарием.

### 4.4. Основания для отклонения/переноса ошибки

Product Owner команды имеет право отклонить (не планировать) ошибку в случаях:

- задача не является ошибкой (например, доработка или случай неверного использования функционала). При необходимости может быть создана задача на описание того, как верно использовать функционал, но ошибка закрывается без решения.
- если есть запланированная доработка по массовому решению проблем одной области (ссылка на задачу и срок обязательны). При этом срок решения доработки не должен превышать требуемый срок исправления ошибки (либо требуется согласие координатора).
- если поднятие приоритета ничем не обосновано и не соответствует правилам определения приоритетов.
- если ошибка не воспроизводится на определенной (более свежей) версии. При этом если после обновления на указанную версию ошибка продолжает воспроизводиться, то координатор имеет право на повышение приоритета проблемы до уровня Критический (Critical)
- если ошибка оформлена не по правилам
- если затраты на исправление существенно превышают затраты на сопровождение
- если ошибка связана с работой устаревших браузеров и операционных систем

#### 4.5. Особенности случаи

Если желаемый срок выполнения не может быть обеспечен, то РО предлагает другой разумный срок. В случае крайней необходимости возможно изменение сроков, путем сдвига сроков по другим задачам и ошибкам, но с согласия ответственных лиц. Задача согласования сроков лежит на РО.

Если нет возможности исправить ошибку (например, требуется серьезный рефакторинг или ошибку тяжело воспроизвести), то РО предлагает альтернативные варианты решения, например:

- формирование доработки с указанным сроком выполнения;
- включение дополнительного логирования.

Если координатор проблемы не согласен с решением РО, то должен связаться с ним и обсудить вопрос лично.

### 5. Процесс работы с задачами по совершенствованию ПО

#### 5.1. Ведение задач по совершенствованию ПО

У каждой команды разработки есть свой отдельный технический бэклог с задачами, целью которых является усовершенствование ПО (далее по тексту *технические задачи*). Задачи в этот бэклог попадают следующим образом:

- непосредственно от команды разработки (любые задачи, которые нельзя совместить с доработками или на которые нужно существенно время). Для каждой задачи должно быть объяснение, что даст её выполнение;
- от менеджера по качеству в результате разбора проблем/ошибок.
- Задачи в бэклоге приоритизируются раз в релиз - совместно командой и менеджером по качеству, после чего помещаются менеджером по качеству в общий бэклог команды и планируются к реализации РО.

#### 5.2. Качество решения технических задач

Предпринимаются следующие (дополнительные) меры, которые позволяют обеспечить качество выполняемых технических задач:

- Каждая техническая задача должна сопровождаться описанием реализации, оформленным в confluence (для схем желательно использовать draw.io);



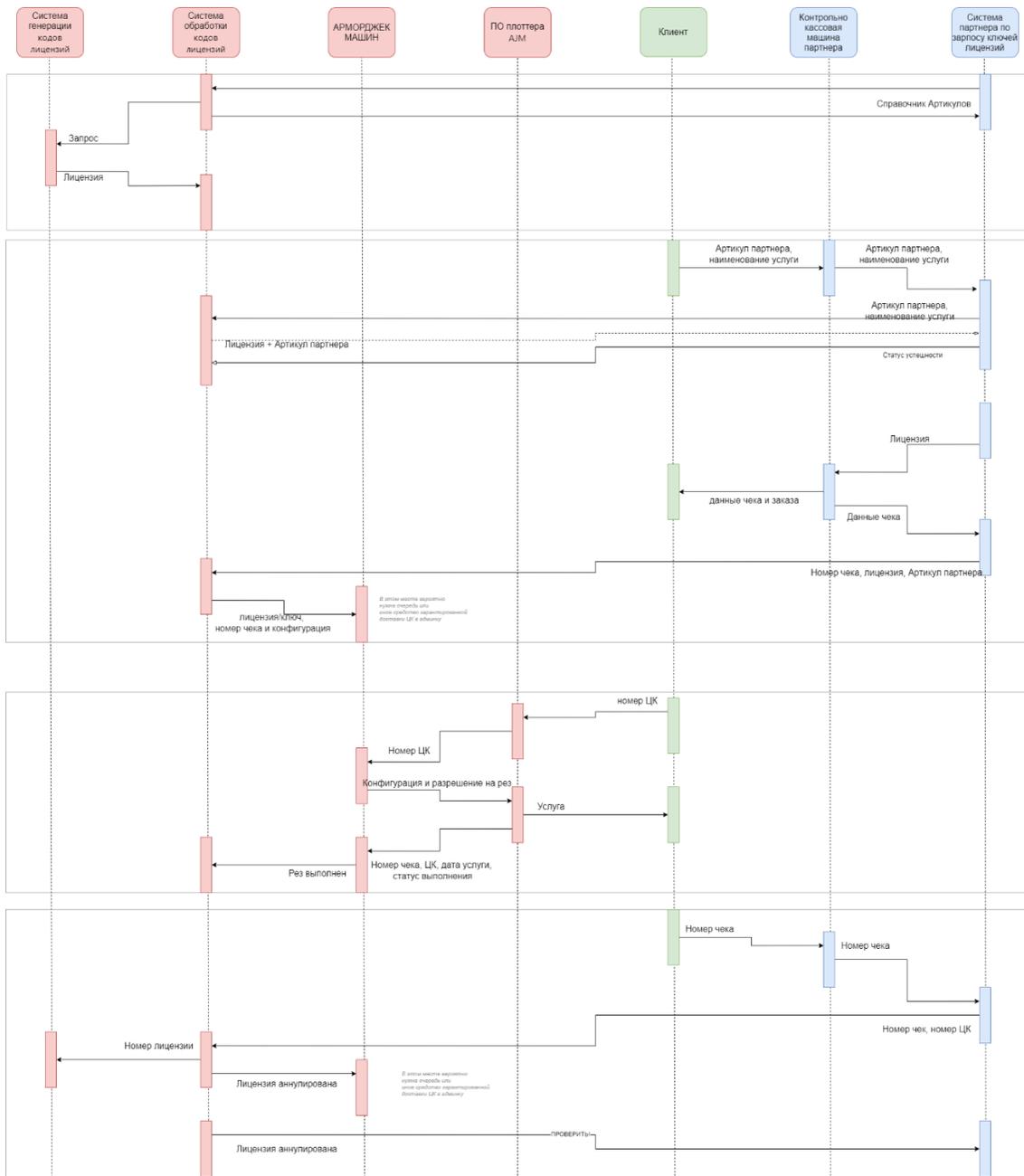


Рисунок 3. Клиентский путь

Клиентский путь разработан так, чтобы быть понятным и легким для пользователя.

В то же время он предлагает дополнительные возможности, которые позволяют пользователю воспользоваться расширенными функциями и персонализированным опытом.



7. Информация о персонале, необходимом для обеспечения поддержки жизненного цикла ПО, в том числе совершенствование ПО, устранение сбойных ситуаций и техподдержку, устранение неисправностей и совершенствование. Работы выполняются собственными силами разработчика, всего задействовано 1 человек-разработчик, 1 тестировщик, 1 специалист техподдержки.
8. Фактический адрес размещения инфраструктуры разработки: 127015, г. Москва, ул. Большая Новодмитровская, дом 14, строение 1.
9. Фактический адрес размещения разработчиков: 127015, г. Москва, ул. Большая Новодмитровская, дом 14, строение 1.\
10. Фактический адрес размещения службы поддержки: 127015, г. Москва, ул. Большая Новодмитровская, дом 14, строение 1.